NOVA SCHOOL OF SCIENCE & TECHNOLOGY

MEEC/MIEEC

Electronics for Micro-Systems

Individual Assignment

Author:

Sofia Margarida Mafra Dias Inácio (58079) sm.inacio@campus.fct.unl.pt

2024/2025 – 1st Semester – DEEC

Contents

List of Figures

List of Tables

1 Problem 1

2 Problem 2

2.1 Identification of the type of filter

The circuit is a two stages filter. The first stage is a first-order high-pass filter, a single RC with a non-inverting OpAmp for buffering and gain. The second stage is a second-order Sallen-Key low-pass filter. Thus, the circuit, shown in Figure 1, is a third-order pass band filter as a whole.

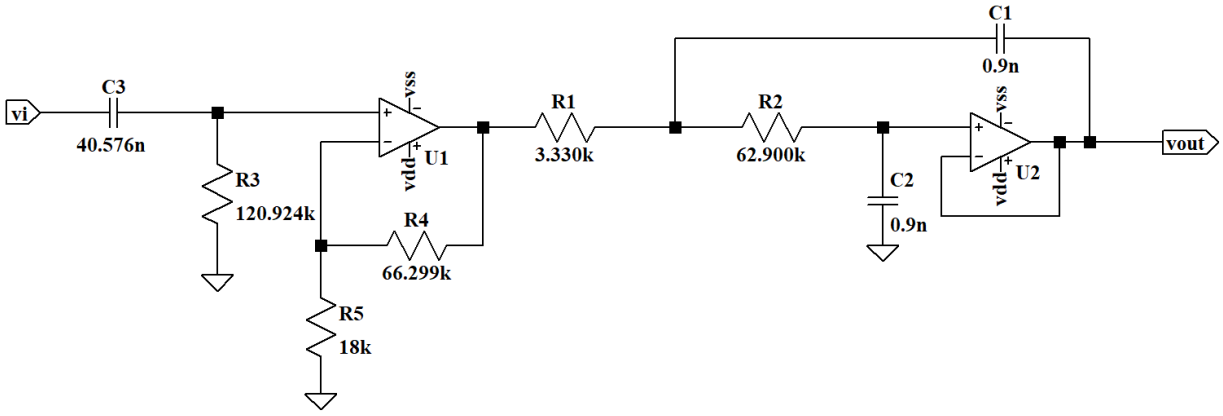

Figure 1: Circuit schematic of the third-order low-pass filter.

Table I: Element values of the circuit.

Element	Value
$R1$	$3.33k\Omega$
$R2$	$62.9k\Omega$
$R3$	$120.92k\Omega$
$R4$	$66.3k\Omega$
$C1$	$0.9nF$
$C2$	$0.9nF$
$C3$	$40.57nF$

2.2 Filter characterization

A Python script was developed to obtain the characteristics of the filter. The script is shown in Listing 1.

```
import scipy as sp
import scipy.signal as sig
import numpy as np
from sympy import *
import matplotlib.pyplot as plt
# Import lambdify and I (imaginary unit) from SymPy
from sympy import symbols, simplify, lambdify, I

# Convert linear value to dB


def lin2dB(val):
    return 20*np.log10(val)


# Define the symbolic variables
vin, v1, vx, vo, s, vos1, vos2 = symbols(
    "V_{in} V_1 V_x V_o s V_{os1} V_{os2}")

# Passa-baixo


na = 58079
n = 1e-9
k = 1000

# Define the components
r1 = 3.33*k
r2 = 62.9*k
r3 = 120.8*k + k*(na/(470000))
r4 = 30*k + k*(na/1600)
r5 = 18*k
c1 = 0.9*n
c2 = 0.9*n
c3 = 40.7*n - n*(na/470000)

# Print the component values
print(f"R1 = {r1/k:.3f} K    ;R2 = {r2/k:.3f} K")
print(f"R3 = {r3/k:.3f} K    ;R4 = {r4/k:.3f} K")
print(f"R5 = {r5/k:.3f} K    ;C1 = {c1*1e9:.3f} n")
print(f"C2 = {c2*1e9:.3f} n    ;C3 = {c3*1e9:.3f} n")
print("\n")

# Define the impedances
z1 = 1/(s*c1)
```

```
z2 = 1/(s*c2)
z3 = 1/(s*c3)


# Stage 1

# V+ = Vin * R3/(R3 + Z3)
# If1 = V+/R5
# V1 - If1*R4 - V+ = 0

vp = vin * r3/(r3+z3)
if1 = vp/r5
s1 = solve(v1 - if1*r4 - vp, v1)

v1 = s1[0]  # output of the first stage

print("First Stage Output:")
pprint(v1)
# print the latex equation
print("\nLatex Eq.: \n"+latex(v1))

# Stage 2

# Vx = V1 - Vr1
# Vo = Z2/(Z2+R2) * Vx

saux = solve(vx*z2/(z2+r2) - vo, vx)
# pprint(saux)

vx = saux[0]

# vo = (vx - v1)/r1 + (vx)/(r2+z2) + (vx-vo)/z1
s2 = solve(
    (vx - v1)/r1 + (vx)/(r2+z2) + (vx-vo)/z1, vo)  # vo

vo = s2[0]  # output of the second stage

# Transfer function
Fs = simplify(vo / vin)

print("Filter Transfer Function: ")
pprint(Fs)

# print the latex equation
print("\nLatex Eq.: \n"+latex(Fs))
```

```
num, den = fraction(Fs)
z = solve(num, s)
p = solve(den, s)

print("Filter Zeros: ")
print(z)
print("Filter Poles: ")
print(p)
print("Frequency:")
for n in p:
    print(f"{abs(n)/(2*np.pi)}")

# Create a function for the transfer function
Fs_numeric = lambdify(s, Fs, 'numpy')

pmax = 6  # Maximum power of 10
pmin = -1  # Minimum power of 10

w = 2 * np.pi * np.logspace(pmin, pmax, 500)  # Frequency range

s_values = 1j * w  # Convert the frequency to the Laplace domain
h = Fs_numeric(s_values)  # Calculate the frequency response

# Find the frequency where the magnitude is 16.9 dB because it is
    the cutoff frequency
ind = np.where(np.isclose(20*np.log10(abs(h)), 16.9, atol=0.05))

print(ind)

# Plot the Bode diagram
plt.figure(figsize=(10, 8))

# Magnitude plot
plt.subplot(2, 1, 1)
plt.semilogx(w/(2*np.pi), 20 * np.log10(abs(h)))
plt.title('Bode Diagram')
plt.ylabel('Magnitude (dB)')
plt.grid(True)
plt.xlim(left=1e-1)

# Phase plot
plt.subplot(2, 1, 2)
plt.semilogx(w/(2*np.pi), np.angle(h, deg=True))
```

```
plt.xlabel('Frequency (Hz)')
plt.ylabel('Phase (degrees)')
plt.grid(True)
plt.xlim(left=1e-1)

plt.tight_layout()
plt.show()

# Find the maximum gain and the frequency where it occurs
MaxGain = max(abs(h))
# Find the position of the maximum gain
MaxPos = np.unravel_index(np.argmax(h), h.shape)
print("Max Gain: ", end="")
print(lin2dB(MaxGain))

# Find the frequency where the maximum gain occurs
print("Freq: ", end="")
print(w[MaxPos]/(2*np.pi))
```

Listing 1: Python script to calculate the filter characteristics.

In Table II are shown the results obtained by the Python script.

Table II: Python script results.

Element	Value
f_{cL}	32.44Ω
f_{cH}	2811.43Ω
f_3	53104.75Ω
A_{max}	13.31

The Bode diagram, generated in Python of the filter response is shown in Figure 3.

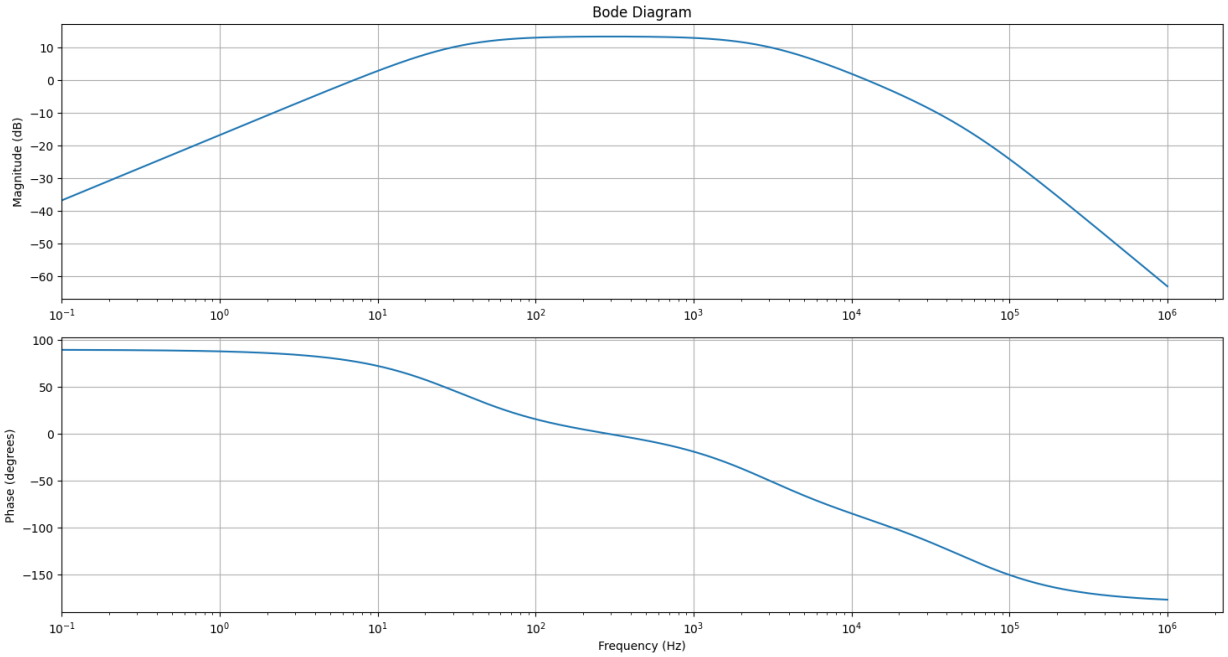

Figure 2: Bode plot of the third-order bandpass filter.

2.3 Filter simulation

In Figure 3 is shown the Bode plot of the filter generated in LTSpice. The low cutoff frequency is around $f_{cL} = 30Hz$ and the high cutoff frequency is around $f_{cH} = 3kHz$. The maximum gain at the passband is around $A_v = 13.3dB$. The poles frequencies are shown in Table III and compared with the theoretical values. The values obtain from the python script are very close to the simulation values.

Table III: Poles on simulation and theoretical values.

Poles	Frequency Simulation	Frequency Theoretical
p_1	$\approx 32.4Hz$	$32.44Hz$
p_2	$\approx 2.8KHz$	$2811.43Hz$
p_3	$\approx 53KHz$	$53.104.76Hz$

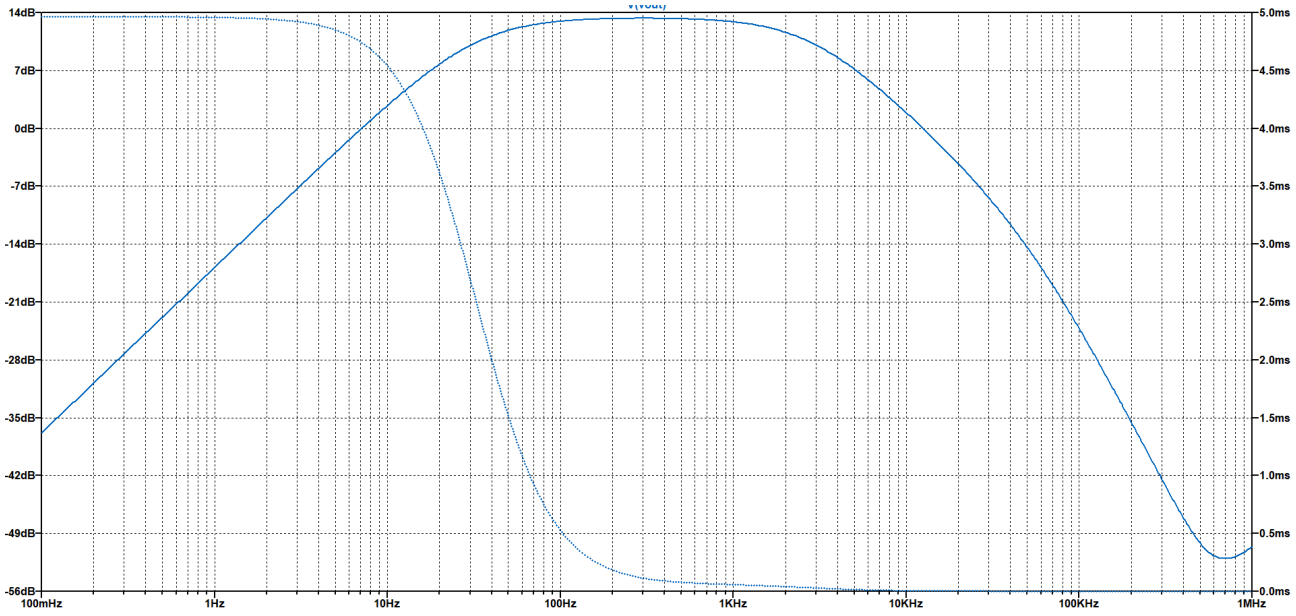

Figure 3: Bode plot of the third-order low-pass filter.

Observing the magnitude response at Figure 3, at low frequencies the magnitude rises and stabilizes at a gain around 13.3dB, this corresponds to the passband gain.

The first pole, the low frequency cut-off pole, is responsible for this gain value, the magnitude stabilizes until the next pole, the high frequency cut-off pole, and after that the third pole increases the attenuation value from $20dB/decade$ to $40dB/decade$.

Observing the group delay, it is constant at the passband, which means that the filter does not introduce any phase distortion at the passband.

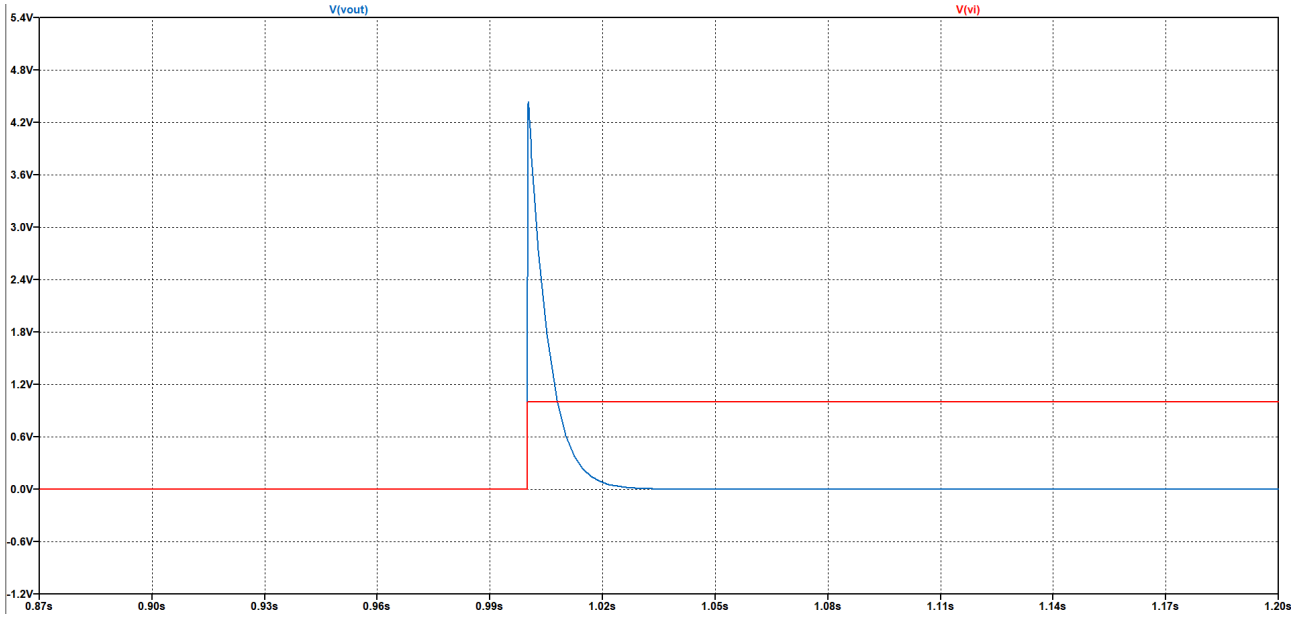

Figure 4: Step response of the third-order low-pass filter.

In Figure 4 is represented the step response of the filter. The step response shows that

the filter is stable and has a fast response time. When the input step rises, the output shows a sharp initial peak, indicating that the filter reacts strongly to sudden changes in the input signal. The response quickly decays back to zero, the filter ability to reject DC component.

2.4 Input offset voltages

Study the impact of the input offset voltages of both OPAMPs. Obtain the expression of the output signal (in the passband region), including the effects of the input offset voltages of both OPAMPs.

Each OPAMP typically has a small input offset voltage, V_{OS} which is an inherent mismatch in the internal transistors. This causes the OPAMP to behave as if a small DC voltage is applied between its inputs, even when the actual input voltages are equal [1].

In Figure 5 is shown the schematic of the filter with the input offset voltages of both OPAMPs.

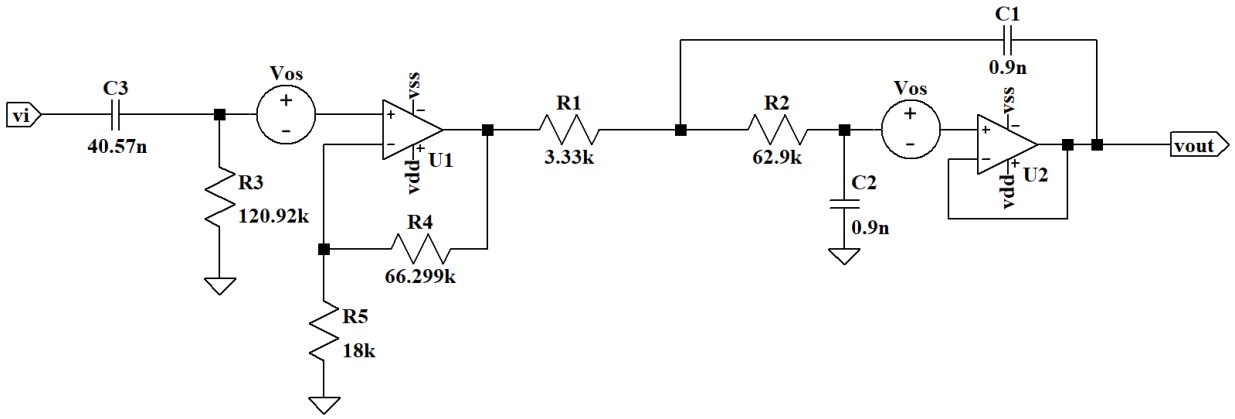

Figure 5: Circuit schematic of the third-order low-pass filter with input offset voltages.

Analyzing the circuit in Figure 5, the output voltage can be calculated using the superposition principle. The output voltage is the sum of the output voltage due to the input offset voltage of the first OPAMP and the output voltage due to the input offset voltage of the second OPAMP.

como vout é igual à tensão no no no ampop ideal o ofsset será vos

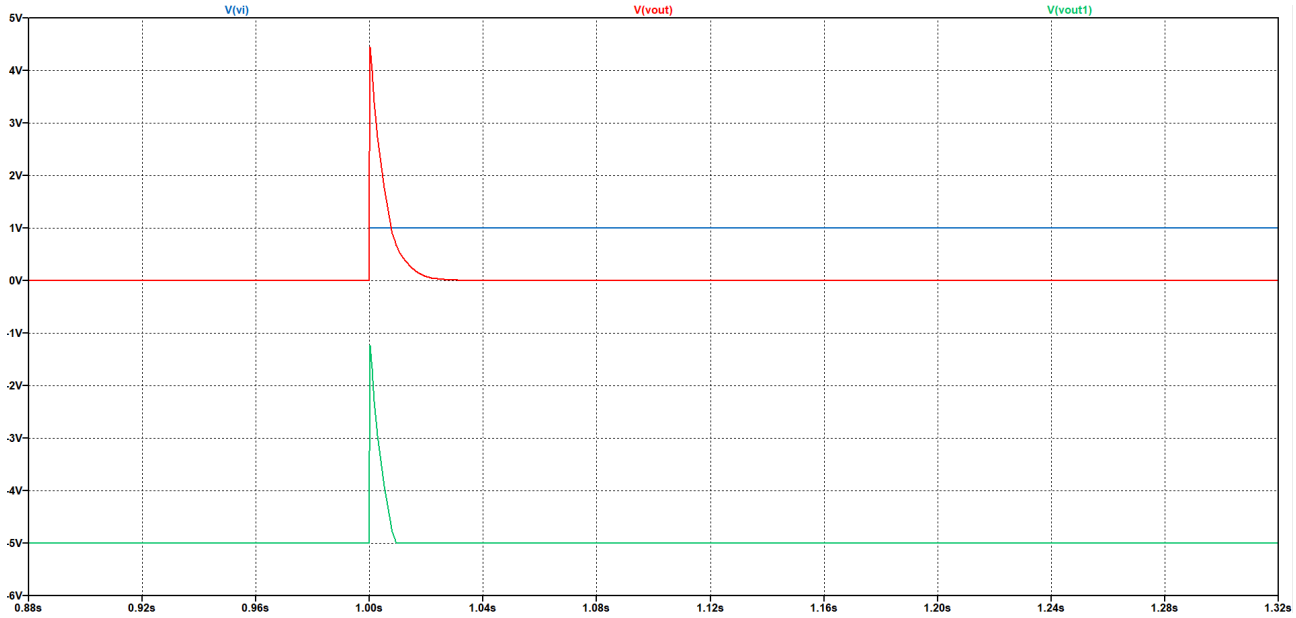

Figure 6: Step response of the third-order low-pass filter with input offset voltages.

Observing the step response in Figure 7, the input offset voltages cause an offset in the output signal, the output signal is not zero when the input signal is zero. The input offset voltages of both OPAMPs causes a DC offset in the output signal.

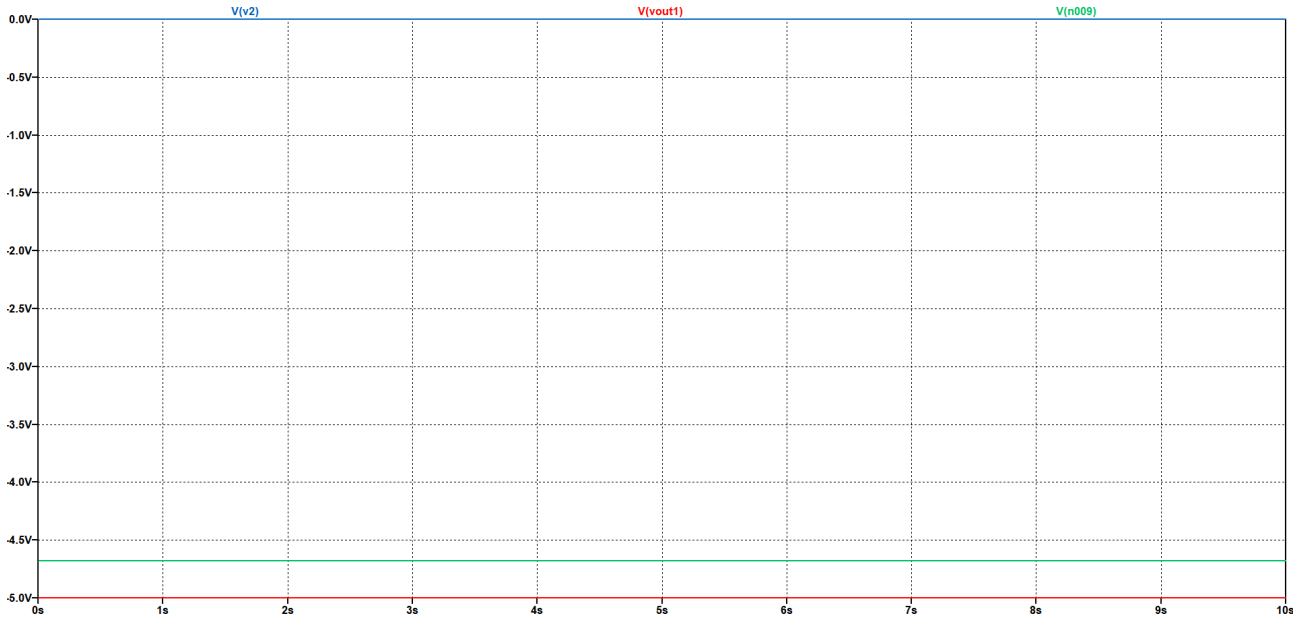

Figure 7: Step response of the third-order low-pass filter with 0V input signal

2.5 GBW Specification

The GBW of an OPAMP defines how its gain decreases with frequency. The closed-loop gain (A_v) at a particular frequency (f) can be calculated using the GBW of the OPAMP and a desired operation frequency, it can be calculated using the following equation 1:

$$A_v = \frac{GBW}{f} \tag{1}$$

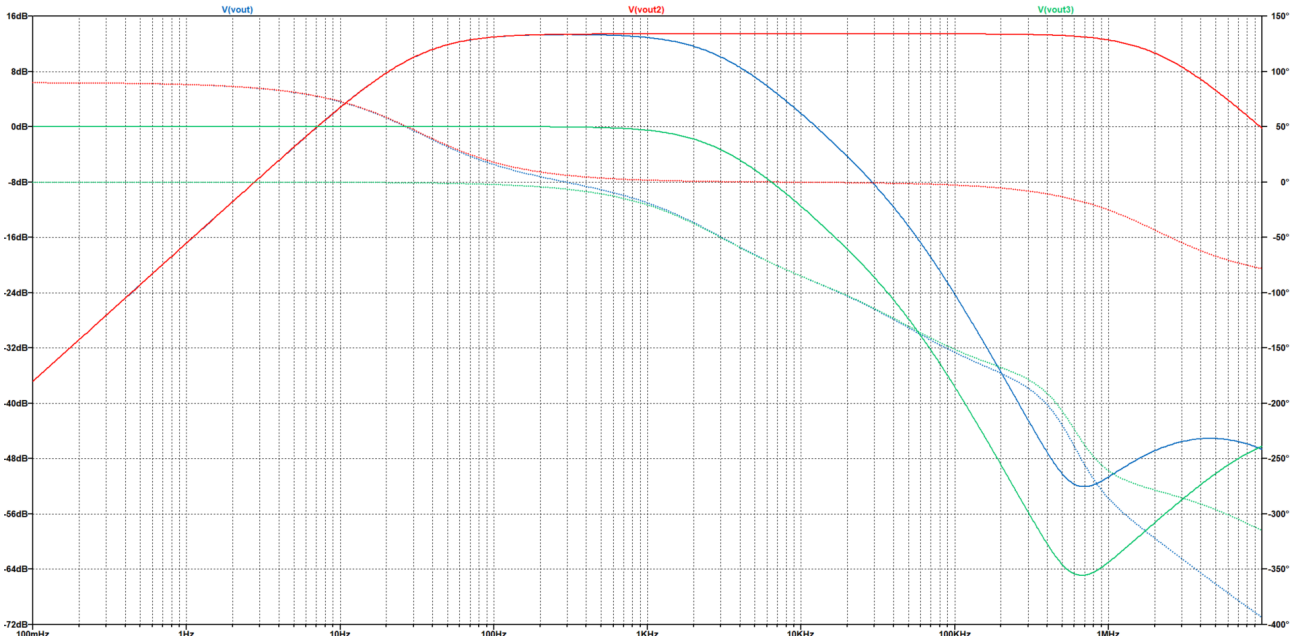

Figure 8: Bode plot of the filter stages and the filter as a whole.

Observing the Figure 8, the gain of the filter is introduced by the first stage of the filter, the second stage is a unity gain buffer. The gain of the first stage is around $13.3dB$. Thus, the first stage needs a higher GBW than the second stage.

The GBW of the first stage is given by:

$$GBW = A_v \times f = 10^{(13.3/20)} \times 3kHz = 12.9kHz \tag{2}$$

The GBW of the second stage is given by:

$$GBW = A_v \times f = 1 \times 3kHz = 3kHz \tag{3}$$

For this filter, the minimum GBW needed for the OPAMP is 12.9kHz. Hence the OPAMP with a GBW of 10kHz is not adequate for the filter.

2.6 Rail-to-rail capability

The maximum signal excursion at the output of an OPAMP is limited. Considering that the OPAMP has a rail-to-rail capability at the output, determine the maximum input signal amplitude below which the output does not saturate.

2.7 OpAmp Selection

Select a commercial OPAMP that fits the filter specifications. Explain which OPAMP parameters/criteria were considered in your decision. For example: GBW?, supply voltage?, . . . , price?, etc.

2.8 Signal to Noise Ratio

Considerer an input sinusoidal input with 10 mV of amplitude and with a frequency inside the pass band of the filter. Obtain the signal to noise ratio at the output node. Consult the input referred noise value of the opamp indicated in the moodle page.

2.9 Single supply version

Modify the circuit of Figure 2 to operate with only one power supply, i.e., propose a single supply version.

3 Problem 6

3.1 Pressure sensor - Principle of operation

A common way of designing a pressure sensor is by using a Wheatstone bridge. The bridge is made up of four resistors sensible to mechanical stress.

One implementation of this sensor is shown in Figure 9. When pressure is applied each resistor either compress or stretch. These sensors are design in order to the opposite resistors in the bridge change their resistance accordingly, i.e., in the example of Figure 9, $R - \triangle R$ decrease by compression, $R + \triangle R$ increase by stretching [2]. This change in resistance yields a change in the differential voltage, $V0$, which is proportional to the applied pressure.

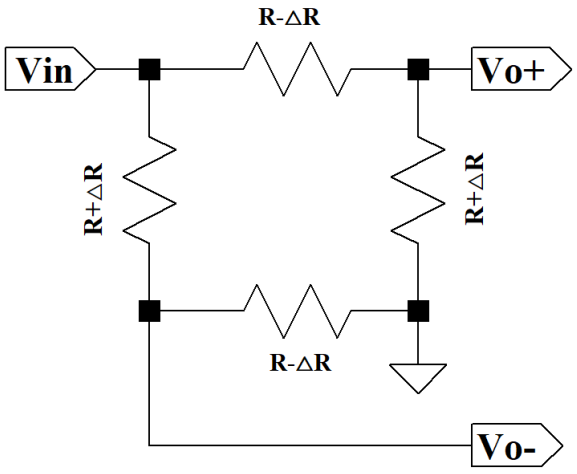

Figure 9: Pressure sensor Wheatstone bridge implementation.

When the bridge is connected to a voltage source and the output voltage is measured across the bridge it is given by the Equation 4:

$$V_0 = V_{in} \cdot \frac{\triangle R}{R} \tag{4}$$

Where V_{in} is the input voltage, $\triangle R$ is the change in resistance and R is the initial resistance of the resistors. Since $\triangle R$ is proportional to the force or pressure, Equation 4 can be rewritten as:

$$V_0 = V_{in} \cdot F \cdot S \tag{5}$$

Where F is the force or pressure and S is the sensitivity of the sensor in mV per volt of excitation with full scale input, specified by the sensor manufacture [3].

3.2 Drip Irrigation System with Pressure Regulation

This system automates the delivery of water to plants through a smart drip irrigation system with controlled pressure. It ensures optimal water flow and avoids overwatering or under-watering by maintaining a target pressure range of $10 - 30psi$, typical for drip irrigation systems. The system uses a pressure sensor, microcontroller, and proportional solenoid valve to regulate the flow based on real-time pressure readings. Additionally, the system features a soil moisture sensor enhancing system's efficiency. In Figure 10 is shown a block diagram of the system.

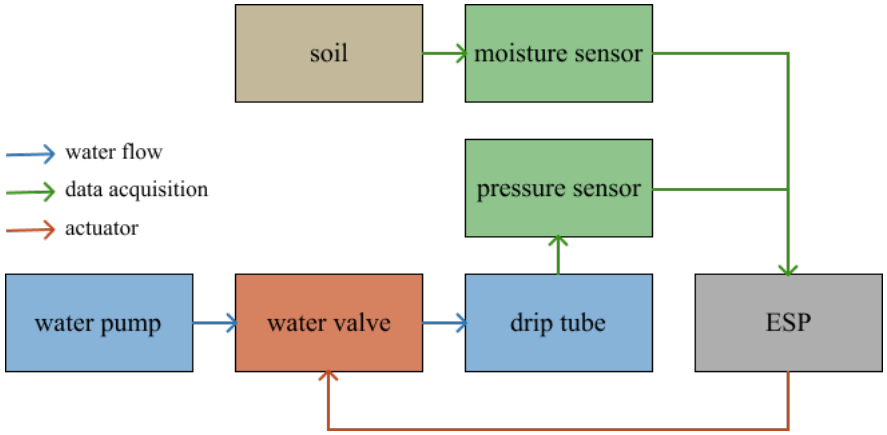

Figure 10: Block diagram of the drip irrigation system.

3.3 System design

3.3.1 Selected Pressure Sensor and Components

- Pressure Sensor: Honeywell PC26 Series BD (0-30 psi variant), with I^2C interface for pressure readings.
- ESP32 Microcontroller: Monitors pressure and controls the solenoid valve while integrating with additional smart features like soil moisture sensing.
- Proportional Solenoid Valve: Adjusts the water flow based on the ESP32's PWM signal to regulate pressure.
- Soil Moisture Sensor: Optional add-on for monitoring soil conditions to further optimize irrigation.

- Water Pump: Ensures consistent pressure in the system.
- 12V DC Power Supply: Powers the pump, valve, ESP32 and sensors.

In Table IV is shown the components and connections of the system.

Table IV: Components and Connections

Component	Connection	Power
Honeywell PC26 Series BD (30 psi) [4]	ESP - I2C Pin	12 VDC
ASCO Series 226 Solenoid Valve [5]	ESP - PWM Pin	12 VDC
ESP32 [6]	N/A	Voltage regulator 12-5 VDC
Soil Moisture Sensor [7]	ESP - I2C Pin	ESP 5VDC
Water Pump	N/A	12 VDC

3.3.2 System Working Principles

Pressure Feedback Loop:
→ Read pressure sensor data via I2C.
→ Compare the pressure reading to the desired range.
→ If below range, open the valve by increasing the PWM duty cycle.
→ If above range, restrict the valve by reducing the PWM signal.

Soil-Based Decision:
→ Read soil moisture levels.
→ If the soil moisture is below a threshold, activate the irrigation cycle.
→ Otherwise, keep the system idle.

3.3.3 Project Implementation

1. Hardware: Connection of the components according to Table IV.
2. Signal Conditioning: Adjustment of the output values of the sensors to the ESP32 input range with an Analog Front-End if needed.
3. Software: Initialization of the system, pressure feedback loop, and soil-based decision-making.
4. Testing and Calibration: Calibration of the pressure sensor and solenoid valve, testing soil moisture thresholds, and the irrigation flow.

FALTA DEFINIR OS RANGES DE ATUAÇÃO DO SISTEMA 10-30 MT VAGO

References

[1] T. Instruments, "Dc parameters: Input offset voltage (vos)," pdf-doc, 2023.

[2] ——, "Design tips for a resistive-bridge pressure sensor in industrial process-control systems," pdf-doc, 2015.

[3] ——, "Signal conditioning wheatstone resistive bridge sensors," pdf-doc, 1999.

[4] Honeywell, "26pc series low pressure sensors, comp/unamp, 0.5 psi to 250 psi datasheet," pdf-datasheet, 2024.

[5] ASCO, "Asco series 226 proportional inline miniature solenoid valves datasheet," pdf-datasheet, 2023.

[6] E. Systems, "Esp32 series datasheet," https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf, 2024.

[7] Dynamax, "Wet150 datasheet," pdf-datasheet, 2024.